

Лекция 10

Распределенные вычислительные системы

Ефимов Александр Владимирович
E-mail: alexandr.v.efimov@sibguti.ru

Курс «Архитектура вычислительных систем»
СибГУТИ, 2020

Понятие о ВС

Вычислительная система (ВС) – совокупность взаимосвязанных и одновременно функционирующих аппаратно-программных вычислителей, которая способна не только реализовать (параллельный) процесс решения сложной задачи, но и априори и в процессе работы автоматически настраиваться и перестраиваться с целью достижения адекватности между своей структурно-функциональной организацией и структурой и характеристиками решаемой задачи.

Распределенная ВС характеризуется тем, что все её аппаратные и программные ресурсы логически и технически распределены между вычислителями. Нет общих ресурсов.

Модель коллектива вычислителей

Вычислительная система – средство обработки информации, базирующееся на модели коллектива вычислителей

$$S = \langle H, A \rangle$$

H – конструкция

A – алгоритм работы коллектива вычислителей.

$$H = \langle C, G \rangle$$

$C = \{c_i\}$ – множество вычислителей c_i , $i = \overline{0, N-1}$

N – мощность множества C

G – описание макроструктуры коллектива вычислителей, т.е. структуры сети связей между вычислителями $c_i \in C$ (или структура коллектива)

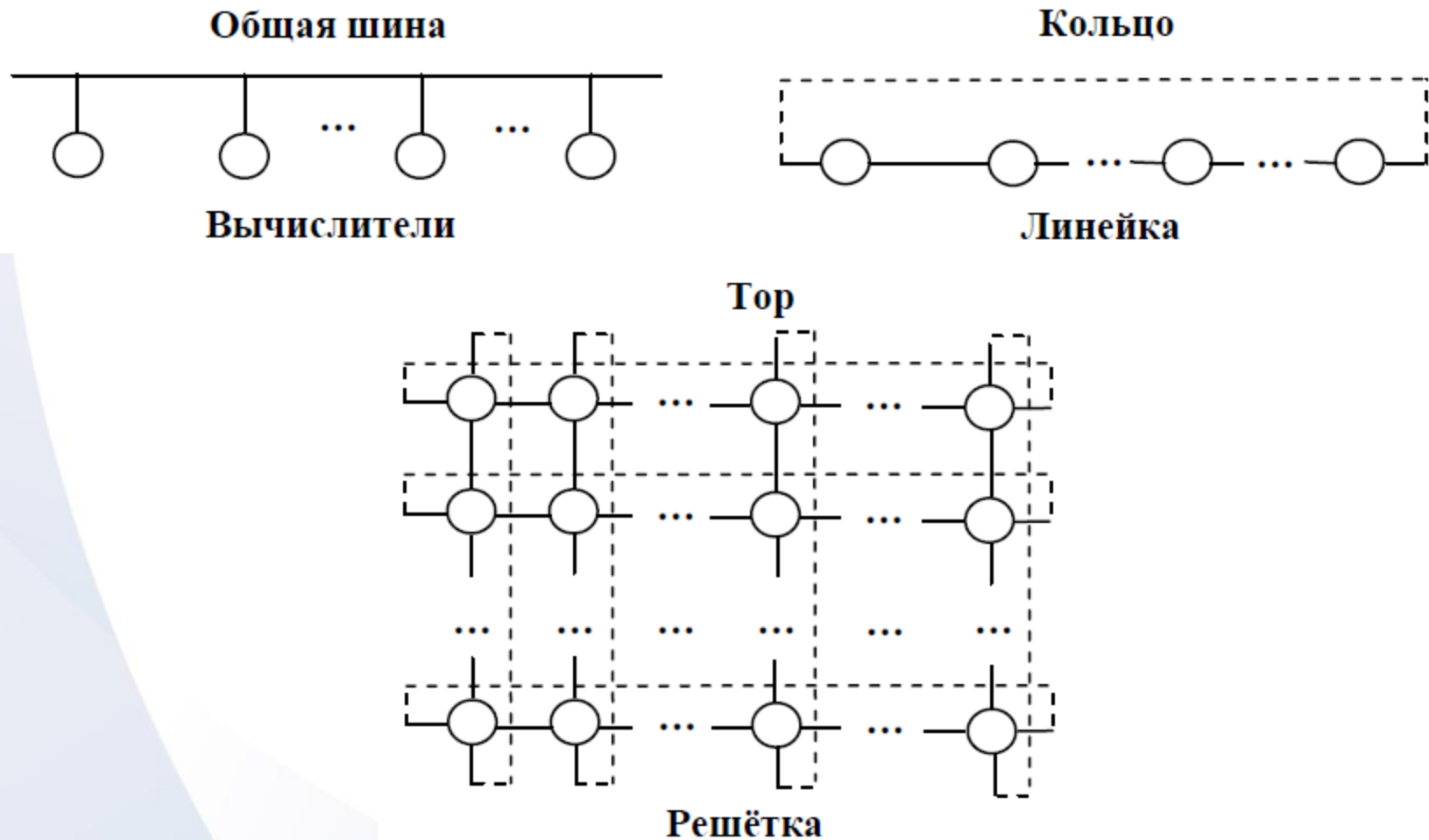
Архитектурные принципы коллектива вычислителей

- *Параллелизм при обработке информации* - параллельное выполнение операций на множестве S вычислителей, взаимодействующих через связи структуры G ;
- *Программируемость структуры* - настраиваемость структуры G сети связей между вычислителями, достигаемая программными средствами;
- *Однородность конструкции H* - однородность вычислителей $s_i \in S$ и структуры G .

Программируемость структуры

Принцип программируемости структуры требует, чтобы в ВС была реализована возможность “хранения” программного описания функциональной структуры и программной ее модификации (перенастройки) с целью достижения адекватности структурам и параметрам решаемых задач.

Простейшие структуры ВС



**Увеличение размерности структуры
повышает структурную надёжность ВС**

Структура ВС (Interconnect)

Сетевая топология — это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (маршрутизаторы), а рёбрам — физические или информационные связи между вершинами.

(Макро)структура коллектива вычислителей представляется *графом* G , вершинам (узлам) которого сопоставлены вычислители $c_i \in C$, а ребрам — линии связи между ними

Коммуникационные сети

- Со статической топологией:

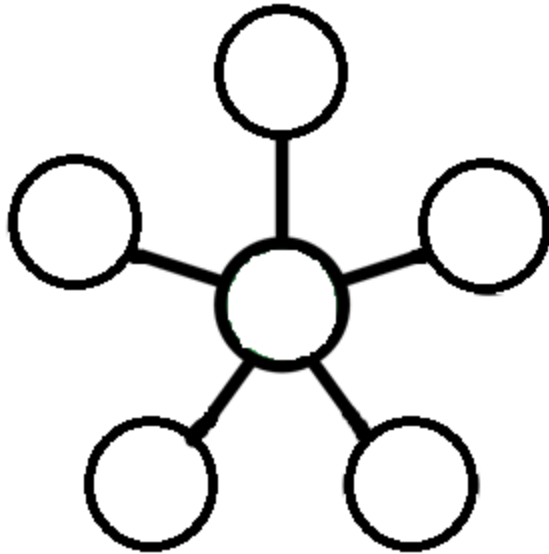
Каждый вычислительный узел имеет сетевой интерфейс (маршрутизатор) с несколькими портами, через который он напрямую соединён с другими узлами

- С динамической топологией

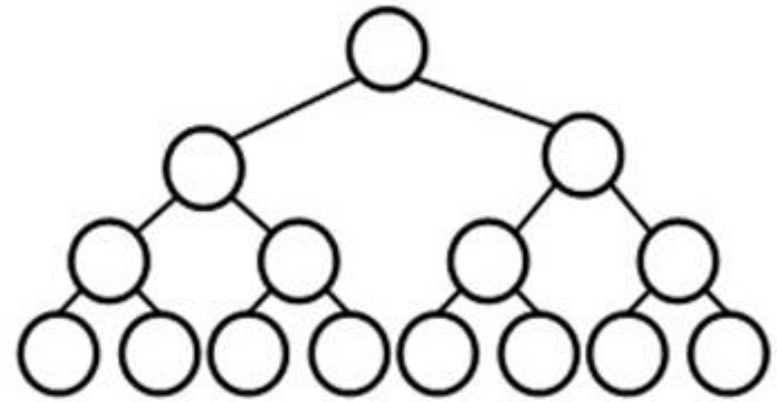
Каждый вычислительный узел имеет сетевой интерфейс с несколькими портами

Порты интерфейсов подключены к коммутаторам, через которые происходит взаимодействие узлов

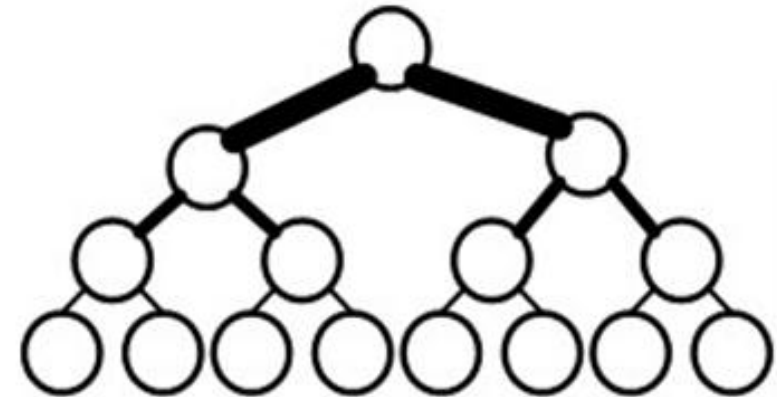
Структура толстого дерева ВС



star

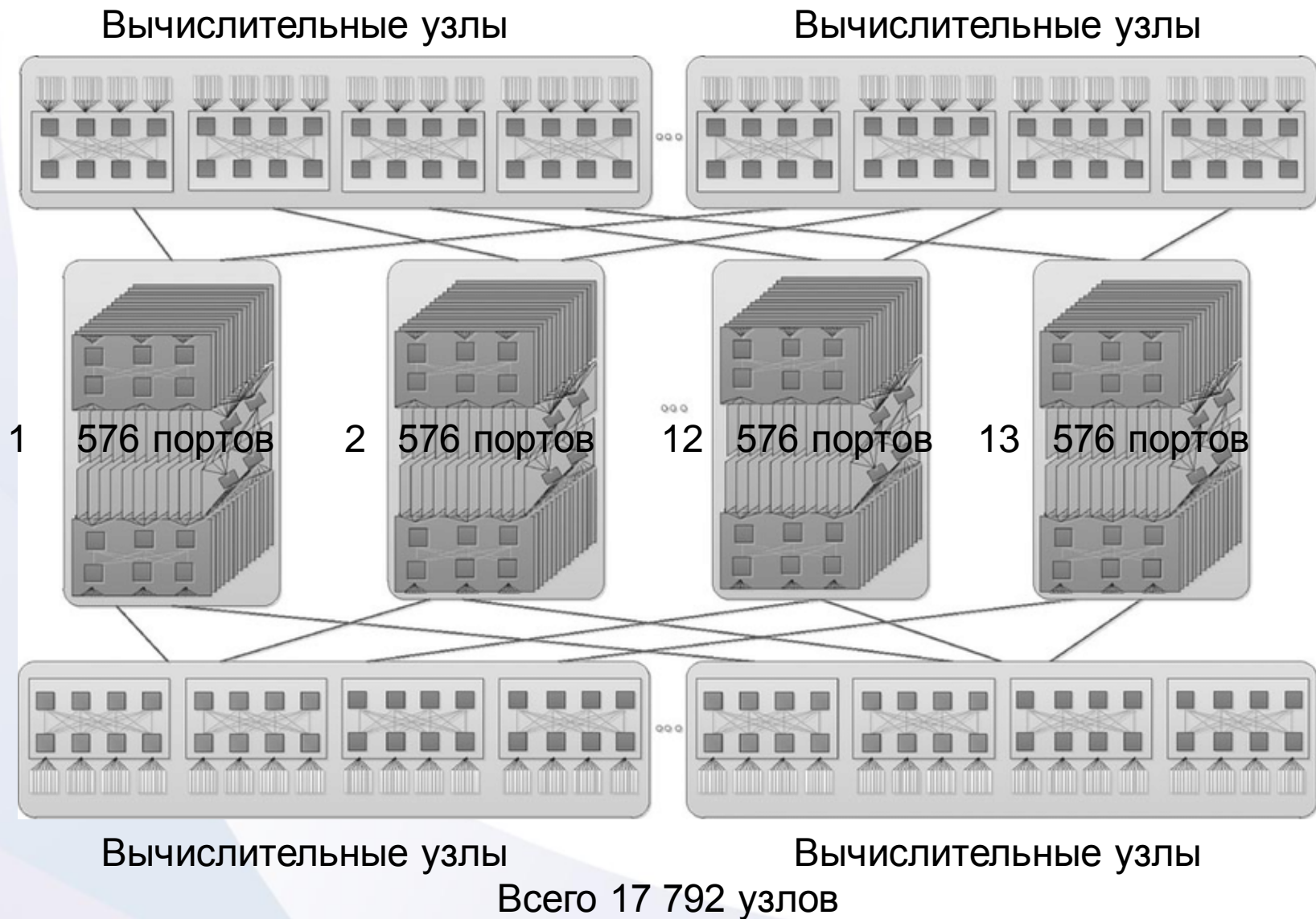


a tree



a fat tree

Tianhe TH Express-2 network architecture



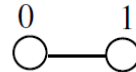
Гиперкубическая структура ВС

➤ Гиперкубы (структуры в виде булевых n -мерных кубов)

$$n = \log_2 N$$

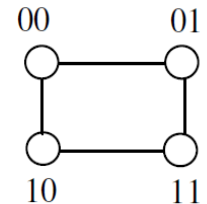
N — количество вершин
 n — число ребер,
выходящих из вершины
(размерность)

Максимальное расстояние
(число ребер) между двумя
вершинами совпадает с
размерностью гиперкуба



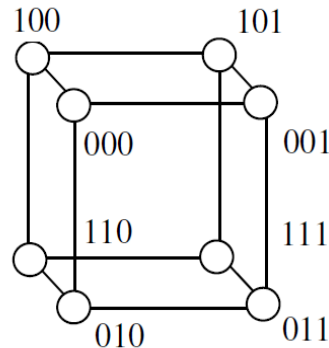
$N = 2, n = 1$

1D — куб



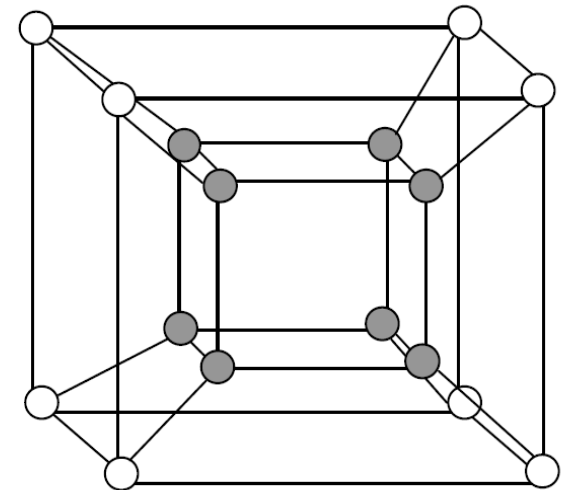
$N = 4, n = 2$

2D — куб



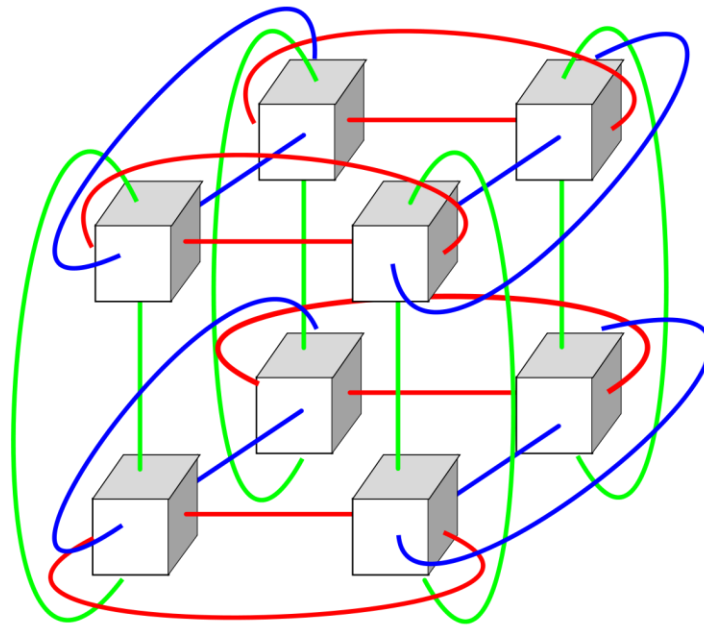
$N = 8, n = 3$

3D — куб



$N = 16, n = 4$, 4D — куб

Структура κ D-тор



3D-тор

Cray Titan 3D torus, IBM Sequoia 5D torus, Fujitsu 6D torus

Структурные характеристики ВС

- *Диаметр структуры* - максимальное расстояние, определённое на множестве кратчайших путей между парами вершин структуры ВС

$$d = \max_{i,j} \{d_{ij}\}$$

- *Средний диаметр структуры*

$$\bar{d} = (N - 1)^{-1} \sum_{l=1}^d l \cdot n_l$$

d_{ij} – расстояние, т.е. минимальное число рёбер, образующих путь из вершины i в вершину j ; $i, j \in \{0, 1, \dots, N - 1\}$;

n_l – число вершин, находящихся на расстоянии l от любой выделенной вершины (однородного) графа G .

Структурные характеристики ВС

➤ Структурная коммутлируемость ВС

$$\mathcal{K}(G, s, s') = \{\mathcal{K}_h(G, s, s')\}, \quad h \in \{1, 2, \dots, [N/2]\}$$

$\mathcal{K}_h(G, s, s')$ - вероятность реализации в системе при заданных структуре G и коэффициентах готовности s и s' соответственно одной ЭМ и линии связи h одновременных непересекающихся межмашинных взаимодействий

➤ Структурная живучесть ВС

$$\mathcal{L}(G, s, s') = \{\mathcal{L}_r(G, s, s')\}, \quad r \in E_2^N = \{2, 3, \dots, N\}$$

$\mathcal{L}_r(G, s, s')$ - вероятность существования подсистемы ранга r (т.е. подмножества из r работоспособных ЭМ, связность которых устанавливается через работоспособные линии связи) при заданных структуре G , коэффициентах готовности s и s' элементарной машины и линии связи соответственно

Структурные характеристики ВС

Бисекционная пропускная способность (bisection bandwidth) – суммарная пропускная способность каналов связи между двумя непересекающимися подмножествами машин системы (для худшего разбиения, минимальное значение)

Аппаратная сложность – число простейших коммутаторов (2×2 , $n \times n$) и каналов связи, необходимых для построения составного коммутатора сети

Метрическая сложность – максимальная длина линии связи, требуемая для реализации выбранной топологии в трехмерном пространстве

Структура ВС

Характеристика	Тип структуры ВС					
	Полный граф	Линейка	Кольцо	2D-решетка	2D-тор	Гиперкуб
Диаметр	1	$N-1$	$[N/2]$	$2(\sqrt{N} - 1)$	$2[\sqrt{N}/2]$	$\log_2 N$
Кол-во рёбер	$N(N-1)/2$	$N-1$	N	$2(N - \sqrt{N})$	$2N$	$(N \log_2 N)/2$

Требования, предъявляемые к структурам ВС

- ✓ *Простота вложения параллельного алгоритма решения сложной задачи в структуру ВС*

Структура ВС должна быть адекватна достаточно широкому классу решаемых задач; настройка проблемно-ориентированных виртуальных конфигураций и реализация основных схем обмена информацией между ЭМ не должны быть связаны со значительными накладными расходами

- ✓ *Удобство адресации элементарных машин и «переноса» подсистем в пределах ВС*

Вычислительная система должна предоставлять возможность пользователям создавать параллельные программы с виртуальными адресами ЭМ. Следовательно, структура ВС должна позволять реализовать простейший «механизм» преобразования виртуальных адресов ЭМ в реальные (физические) адреса машин ВС

Требования, предъявляемые к структурам ВС

- ✓ *Осуществимость принципа близкодействия и минимума задержек при межмашинных передачах информации в ВС*

Принцип близкодействия предопределяет реализацию обменов информацией между «удалёнными» друг от друга ЭМ через промежуточные машины системы. Следовательно, в условиях ограниченности числа связей у каждой ЭМ структура должна обеспечивать минимум задержек при «транзитных» передачах информации

- ✓ *Масштабируемость и большемасштабность структуры ВС*

Для формирования конфигураций ВС с заданной эффективностью требуется, чтобы структура обладала способностью к наращиванию и сокращению числа вершин (машин). Изменение числа ЭМ в ВС не должно приводить к коренным перекоммутациям между машинами и (или) к необходимости изменения числа связей для любых ЭМ

Требования, предъявляемые к структурам ВС

✓ *Коммутируемость структуры ВС*

ВС должна быть приспособлена к реализации групповых межмашинных обменов информацией. Следовательно, структура ВС должна обладать способностью осуществлять заданное число одновременных непересекающихся взаимодействий между элементарными машинами

✓ *Живучесть структуры ВС*

Важным требованием к ВС в целом является обеспечение работоспособности при отказе её компонентов или даже подсистем

✓ *Технологичность структур ВС*

Структура сети межмашинных связей ВС не должна предъявлять особых требований к элементной базе, к технологии изготовления микропроцессорных БИС. Системы должны быть восприимчивы к массовой технологии, их «вычислительное ядро» должно формироваться из массовых микропроцессорных БИС. Последнее позволит достичь приемлемых значений технико-экономических показателей ВС



Анализ путей удовлетворения перечисленным требованиям приводит к безальтернативному выбору *однородных* (или регулярных, т.е. описываемых однородными графами) структур для формирования вычислительных систем

Перспективные структуры ВС

➤ Циркулянтные структуры (D_n -графы)

D_n -граф или циркулянтная структура есть граф G вида:

$\{N; \omega_1, \omega_2, \dots, \omega_n\}$, в котором:

N – число вершин или порядок графа;

вершины помечены целыми числами i по модулю N , следовательно,
 $i \in \{0, 1, \dots, N-1\}$;

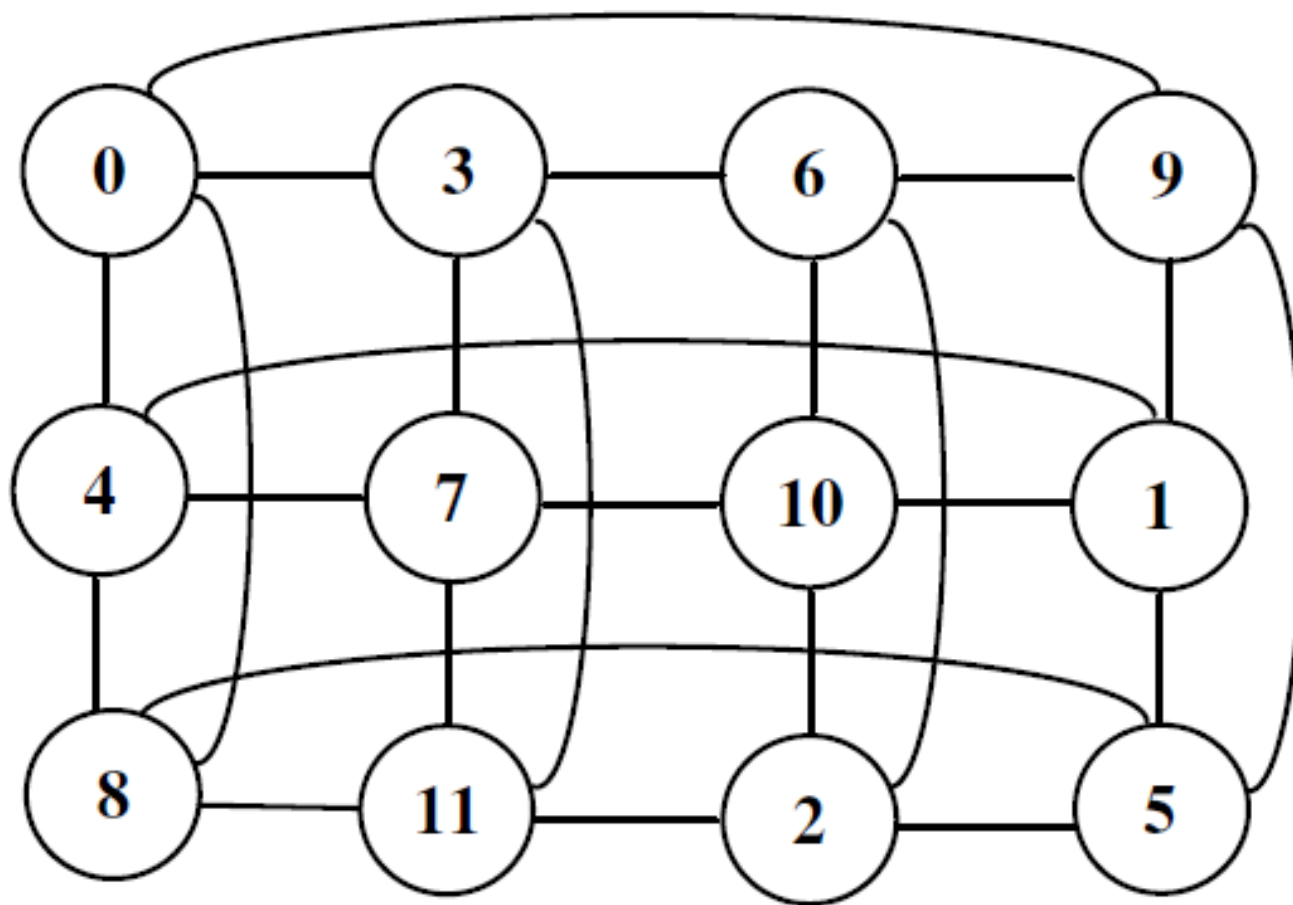
вершина i соединена ребром (или является смежной) с вершинами
 $i \pm \omega_1, i \pm \omega_2, \dots, i \pm \omega_n \pmod{N}$;

$\{\omega_1, \omega_2, \dots, \omega_n\}$ – множество целых чисел, называемых образующими, таких, что $0 < \omega_1 < \omega_2 < \dots < \omega_n < (N + 1) / 2$, а для чисел $N; \omega_1, \omega_2, \dots, \omega_n$ наибольшим общим делителем является 1;

n – размерность графа;

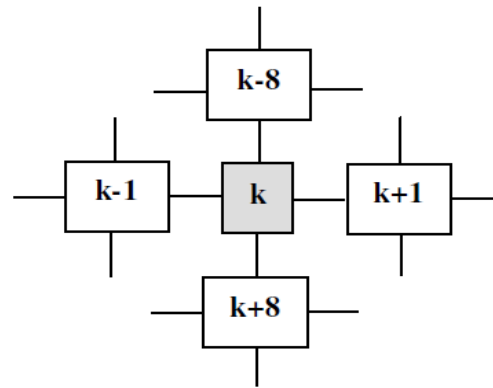
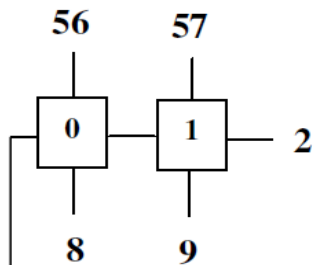
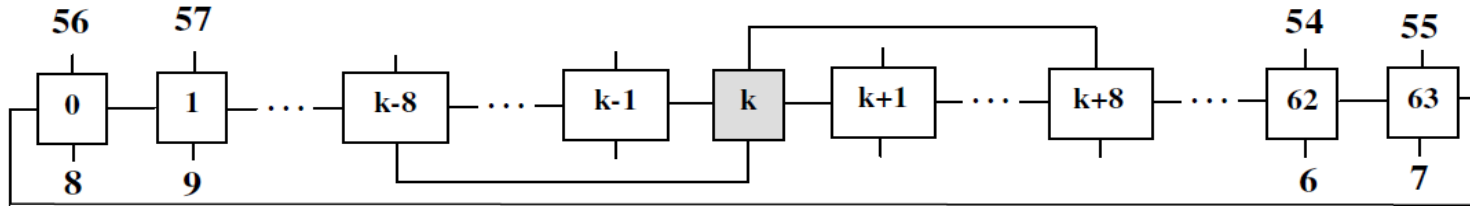
$2n$ – степень вершины в графе.

D_2 -граф или двумерный циркулянт вида:
 $\{12;3,4\}$

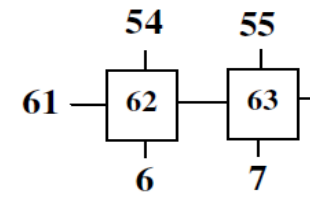


Циркулянтные структуры (D_n -графы)

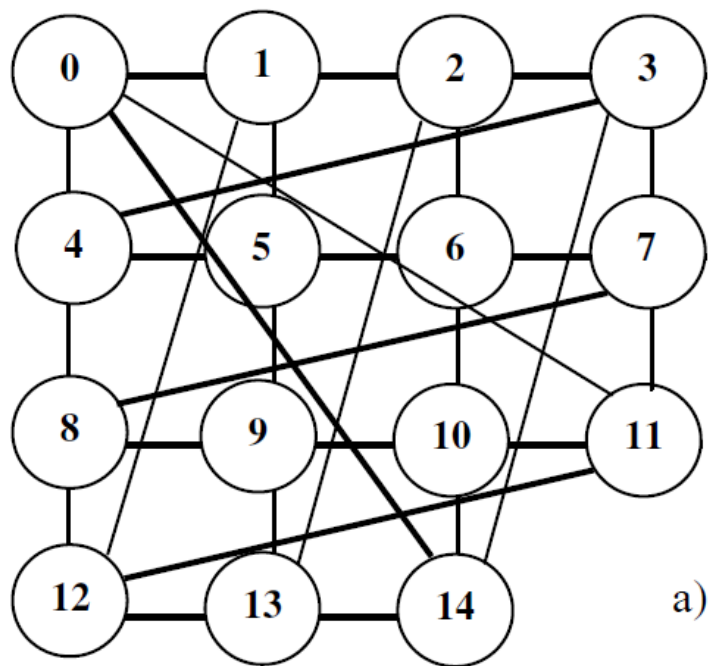
- Циркулянтные структуры $\{N; 1, \omega_2\}$ широко внедрены в практику вычислительных систем (матричные ВС)



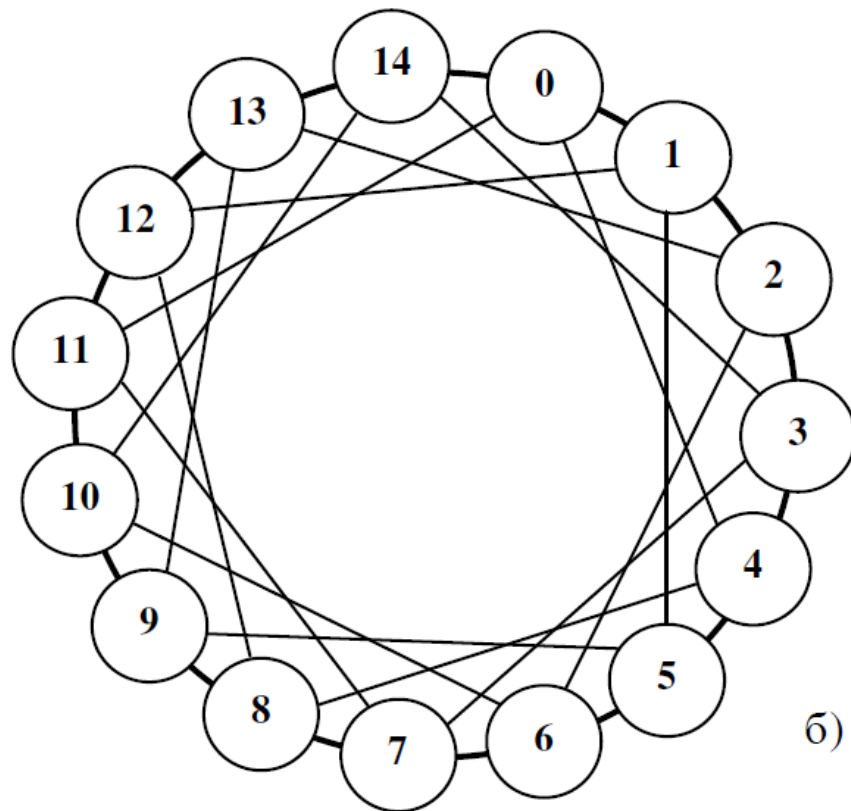
Циркулянт $\{64; 1, 8\}$, отражающий структуру квадранта вычислительной системы ILLIAC-IV.



Циркулянтные структуры (D_n -графы)



a)



б)

D_2 -граф: $\{15;1,4\}$: а) – двумерная матрица, б) – хордовое кольцо

Циркулянтные структуры (D_n -графы)

- ✓ Целые числа $i \in \{0, 1, 2, \dots, N-1\}$, отмечающие вершины D_n -графа, называют *адресами*.
- ✓ Адресация вершин в таких структурах называется *диофантовой* (в честь древнегреческого математика из Александрии Диофанта, Diophantos, 3 в.).
- ✓ В циркулянтных структурах при полном переносе какой-либо подструктуры (всех вершин подструктуры на одно и то же расстояние в одном из направлений) сохраняются все её свойства и адресация вершин.
- ✓ Следовательно, при диофантовой адресации элементарных машин ВС можно простыми средствами реконфигурации осуществить виртуальную адресацию вершин-машин и, следовательно,
 - 1) создавать отказоустойчивые параллельные программы,
 - 2) реализовывать мультипрограммные режимы обработки информации,
 - 3) исключать отказавшие вершины-машины из подсистем, а значит обеспечить живучесть ВС.

Циркулянтные структуры (D_n -графы)

- Реконфигурация сводится к изменению адресов α у всех машин подсистемы

$$\alpha := [\alpha + (j - i)] \bmod N, \quad \alpha \in \{0, 1, \dots, N - 1\}$$

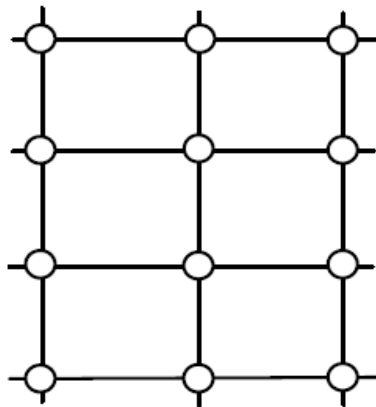
i – номер ЭМ, исключаемой из подсистемы

j – номер машины, включаемый в подсистему

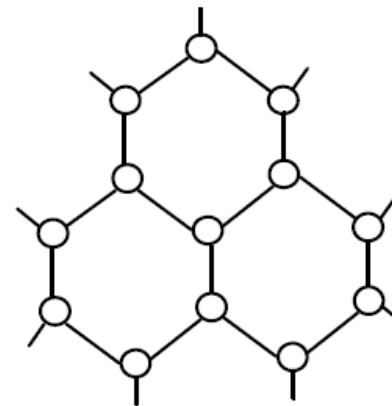
$i, j \in \{0, 1, \dots, N - 1\}$.

$L(N, v, g)$ -графы

- Структуры ВС, допускающих масштабирование (изменение числа машин) без коренной перекоммутации уже имеющихся межмашинных связей
- $L(N, v, g)$ -граф – это неориентированный однородный граф с числом и степенями вершин соответственно N и v и значением обхвата g
- Каждая вершина при $v \geq 3$ входит в не менее v кратчайших простых циклов длиной g (длина кратчайшего цикла в графе называется *обхватом*)
- При $v = 2$ $L(N, v, g)$ -граф является простым циклом с N вершинами



а)



б)

Фрагменты $L(N, v, g)$ -графов:

а) – $v = 4, g = 4$; б) – $v = 3, g = 6$

Анализ и синтез структур ВС

- Расчёт значений структурных показателей ВС
- Получение аналитических выражений для координат вектор-функций структурной коммутируемости ВС и структурной живучести является сложной задачей, разрешимой лишь для частных случаев. Рабочий метод расчёта этих показателей – статистическое моделирование
- *Проблема синтеза структур* заключается в поиске таких графов G^* , которые бы делали реальные (физические) конфигурации ВС максимально приспособленными для программирования (виртуальных) конфигураций

Постановка задачи синтеза структур ВС

- Найти структуру G^* , которая обеспечивала бы максимум координаты вектор-функции структурной живучести

$$\max_G \mathcal{L}_r(G, s, s') = \mathcal{L}_r(G^*) \quad (1)$$

при заданных значениях N, r, v, s, s' .

- Структура G^* , для которой выполняется (1), называется *оптимальной*.
- Проблема синтеза оптимальных структур большемасштабных ВС относится к сложным проблемам, она практически решается при помощи статистического моделирования (методом Монте-Карло) и, следовательно, с использованием мощных вычислительных средств

Оптимальные структуры ВС

- Трудоёмкость поиска G^* можно заметно снизить, если воспользоваться двумя нижеприведенными гипотезами.
- **Гипотеза 1.** Структура G^* , при которой достигается $\mathcal{L}_N(G^*)$ – максимум живучести ВС, обеспечивает и $\mathcal{L}_r(G^*)$ – максимум живучести подсистем ранга $r < N$.
- **Гипотеза 2.** Структура с минимальным (средним) диаметром относится к G^* , т.е. обладает максимальной структурной живучестью.
- Справедливость гипотез подтверждена результатами статистического моделирования структур ВС. Эти гипотезы были высказаны В.Г. Хорошевским в 1970-х годах

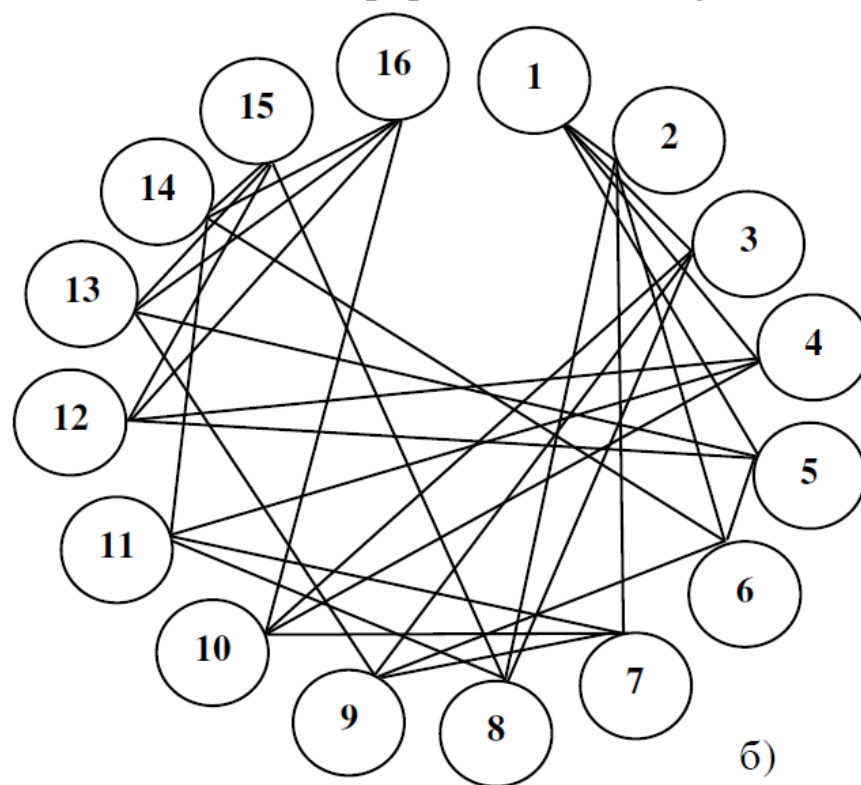
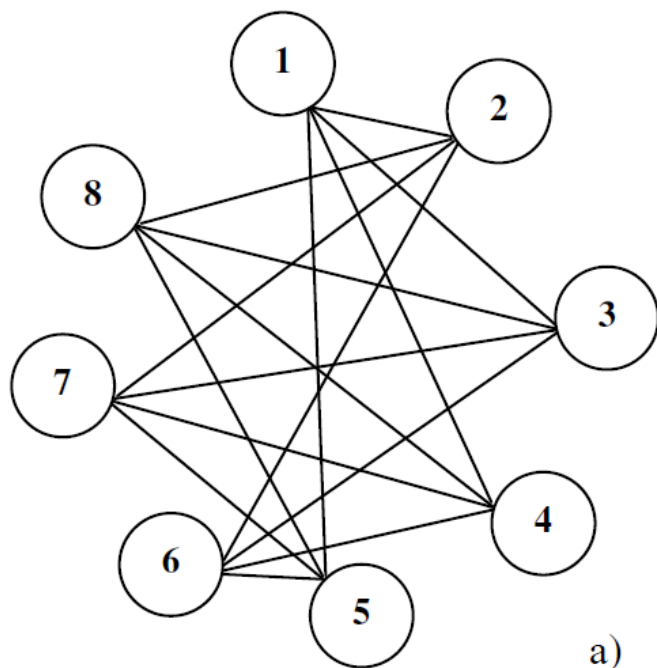
Оптимальные структуры ВС

- *Оптимальными* будем называть структуры G^* , имеющие при заданных порядке N и степени v вершин минимальный диаметр.
- Создание общего алгоритма синтеза оптимальных структур является сложной задачей. Существуют алгоритмы синтеза для конкретных классов графов.
- Для целей практики созданы и пополняются каталоги оптимальных структур.

Фрагмент каталога оптимальных D_n -графов

D_n -граф	N	ω_1	ω_2	ω_3	ω_4	ω_5
D_2 -граф	16	1	6			
	32	1	7			
	64	1	14			
	128	1	15			
	256	1	92			
D_3 -граф	16	1	2	6		
	32	1	4	10		
	50	1	8	12		
	2048	37	116	202		
		48	407	615		
		349	3 90	686		
D_4 -граф	16	1	2	3	4	
	32	1	2	8	13	
	64	1	4	10	17	
D_5 -граф	16	1	2	3	4	5
	32	1	2	3	4	12
	50	1	3	8	16	20
	1024	22	189	253	294	431
		30	133	230	253	485
		6	317	403	425	475

Оптимальные $L(N, v, g)$ -графы



Оптимальные $L(N, v, g)$ -графы:

а) $L(8, 4, 4)$ -граф, $d = 2$, $\bar{d} = 1,43$; б) $L(16, 4, 4)$ -граф, $d = 3$, $\bar{d} = 1,91$

Оптимальные $L(N, v, g)$ -графы

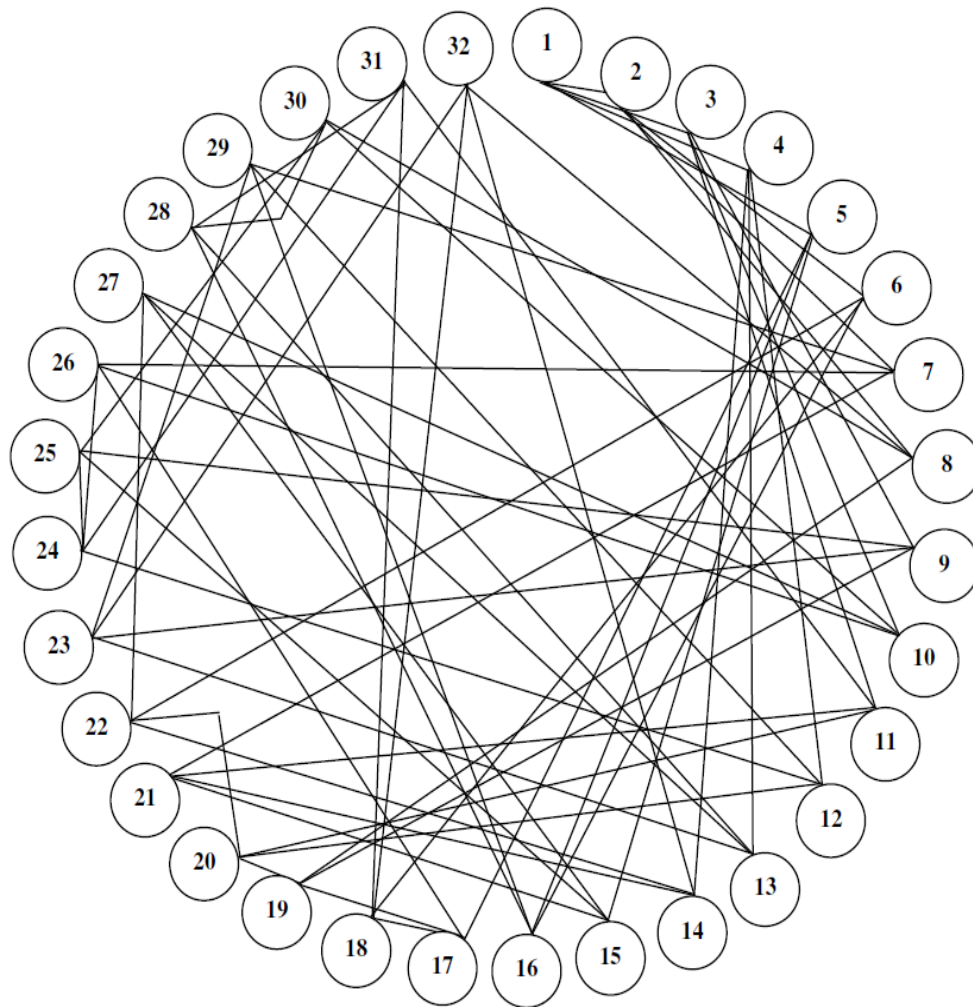
- Графы $L(N, v, g)$ можно описывать в виде *матриц смежности*.
 i, j, k, \dots, l - элементы i -той строки матрицы смежности $L(N, v, g)$ -графа, которые равны 1

$L(32,3,7)$ -граф, $d = 5$, $\bar{d} = 2,94$

1. 2,3,4	9. 4,19,20	17. 8,12,26	25. 13,16,26
2. 1,5,6	10. 4,21,22	18. 8,19,29	26. 17,20,25
3. 1,7,8	11. 5,22,29	19. 9,18,31	27. 14,20,28
4. 1,9,10	12. 5,17,23	20. 9,26,27	28. 21, 27,32
5. 2,11,12	13. 6,24,25	21. 10,23,28	29. 11,18,32
6. 2,13,14	14. 6,15,27	22. 10,11,30	30. 15,22,31
7. 3,15,16	15. 7,14,30	23. 12,21,24	31. 19,24,30
8. 3,17,18	16. 7,25,32	24. 13,23,31	32. 16,28,29

Оптимальный $L(32, 4, 6)$ -граф;

$d = 4, \bar{d} = 2,36$



$L(32,4,6)$ -граф, $d = 4, \bar{d} = 2,36$

- | | | |
|----------------|-----------------|-----------------|
| 1. 2,3,4,5 | 12. 4,20,24,29 | 23. 9,13,29,32 |
| 2. 1,6,7,8 | 13. 4,23,27,28 | 24. 12,25,26,31 |
| 3. 1,9,10,11 | 14. 4,21,22,32 | 25. 9,15,24,30 |
| 4. 1,12,13,14 | 15. 5,21,25,27 | 26. 7,10,17,24 |
| 5. 1,15,16,17 | 16. 5,6,28,29 | 27. 10,13,15,22 |
| 6. 2,16,18,22 | 17. 5,18,19,26 | 28. 13,16,30,31 |
| 7. 2,21,26,29 | 18. 6,17,31,32 | 29. 7,12,16,23 |
| 8. 2,19,30,32 | 19. 8,9,17,20 | 30. 8,10,25,28 |
| 9. 3,19,23,25 | 20. 11,12,19,22 | 31. 11,18,24,28 |
| 10. 3,26,27,30 | 21. 7,11,14,15 | 32. 8,14,18,23 |
| 11. 3,20,21,31 | 22. 6,14,20,27 | |

Сравнительный анализ структур ВС

$N = 2^v$	Гиперкубы		Циркулянты			$\mathcal{L}(N, v, g)$ -графы			
	$v = d$	\bar{d}	v	d	\bar{d}	v	g	d	\bar{d}
64	6	3.0	6	4	2.5	6	6	3	2.29
256	8	4.0	8	4	3.3	8	6	3	2.7
512	9	4.5	8	5	4.02	9	6	3	2.81
1024	10	5.0	10	5	4.04	10	6	4	3.01
2048	11	5.5	10	6	4.70	11	6	4	3.47
4096	12	6.0	12	6	4.68	12	6	4	3.57
8192	13	6.5	12	6	5.34	13	6	4	3.78
16384	14	7.0	14	6	5.38	14	6	4	3.83
32768	15	7.5	14	7	6.09	15	6	4	3.89
65536	16	8.0	16	7	6.12	16	6	5	4.06
131072	17	8.5	16	8	6.73	17	6	5	4.39
262144	18	9.0	18	8	6.75	18	6	5	4.62
1048576	20	10.0	20	8	7.41	20	6	5	4.85
16777216	24	12.0	24	10	8.76	24	6	6	5.56
268435456	28	14.0	28	11	10.15	28	6	6	5.94

Сравнительный анализ структур ВС

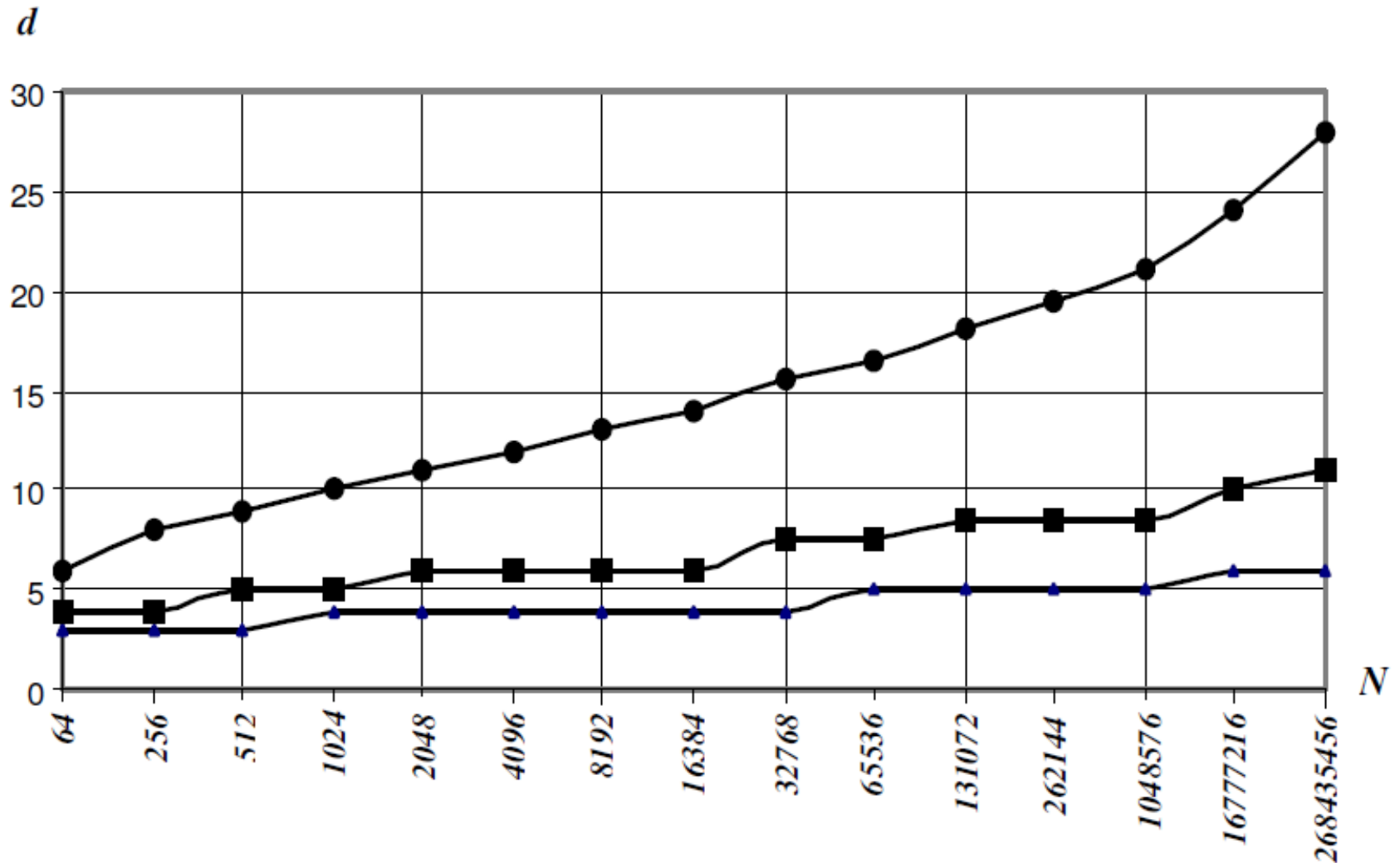


Рис. 7.6 Зависимость диаметра структуры ВС от числа вершин;

—●— — гиперкуб, —■— — D_n -граф, —▲— — $L(N, v.g)$ -граф

Сравнительный анализ структур ВС

- D_n -графы при одинаковых (или даже меньших на единицу) степенях имеют диаметры, которые меньше чем в 3 раза по сравнению с диаметром гиперкубов
- D_n -графы обладают меньшими средними диаметрами по сравнению с гиперкубам
- Рассматриваемые показатели для $L(N, v, g)$ -графов при $g > 4$ самые лучшие: так диаметры для оптимальных $L(N, v, g)$ -графов оцениваются величиной $0,21 * \log_2 N$, в то время как в гиперкубах – $\log_2 N$
- $L(N, v, g)$ -графы характеризуются логарифмической зависимостью диаметров от количества N вершин при фиксированной степени вершин



В вычислительных системах, использующих D_n - и $L(N, v, g)$ -графы, время межмашинных (межпроцессорных) обменов информацией значительно меньше по сравнению с временем гиперкубических ВС



D_n - и $L(N, v, g)$ -графы более перспективны для формирования сетей межмашинных (межпроцессорных) связей в ВС, чем гиперкубы

ВС с программируемой структурой

Вычислительные системы (ВС) с программируемой структурой – это распределенные средства обработки информации, в которых все компоненты (устройство управления, процессор и память) являются распределенными (нет единого функционально и конструктивно реализованного устройства).

Тип архитектуры ВС – MIMD.

Допускается формирование конфигураций с массовым параллелизмом (Scalable Massively Parallel Architecture Computing Systems).

ВС с программируемой структурой

Под ВС с программируемой структурой понимается совокупность элементарных машин, функциональное взаимодействие между которыми осуществляется через программно настраиваемую сеть связи.

ВС с программируемой структурой

Элементарная машина (ЭМ) – это композиция из вычислительного модуля и системного устройства.

Вычислительный модуль (ВМ) служит как для переработки и хранения информации, так и для выполнения функций по управлению системой в целом (аппаратурно-программный комплекс).

Системное устройство (СУ) – это ЛК или КП, предназначенные для взаимодействия данной ЭМ с ближайшими соседними машинами (точнее, с системными устройствами, с которыми имеется непосредственная связь).

Вычислительная система «Минск-222»

Разработана и построена Отделением вычислительной техники Института математики СО АН СССР совместно с Конструкторским бюро завода им. Г.К. Орджоникидзе Министерства радиопромышленности СССР (г. Минск).

Руководитель работ: Э.В. Евреинов.

Основные разработчики:

В.Г. Хорошевский, Б.А. Сидристый, Г.П. Лопато.

Начало работ - 1965 год.

Первый образец 1966 год.

Системы “Минск-222” были смонтированы в нескольких организациях Советского Союза и эксплуатировались более 15 лет.

Архитектурные особенности «Минск-222»

- MIMD-архитектура, распределённость ресурсов;
- параллелизм, однородность, программируемость структуры;
- одномерная (кольцевая) топология;
- масштабируемость: 1 – 16 элементарных машин (ЭМ);
- быстродействие: $\Omega = A \cdot N \cdot \omega$,
 N - число ЭМ,
 ω - быстродействие одной ЭМ,
 $A \geq 1$ - коэффициент (при крупноблочном распараллеливании сложных задач);
- использование промышленных ЭВМ второго поколения.

Характеристики

Быстродействие 5-6тыс. оп/с

Ёмкость оперативной памяти - 8192 слов.

Количество разрядов - 37

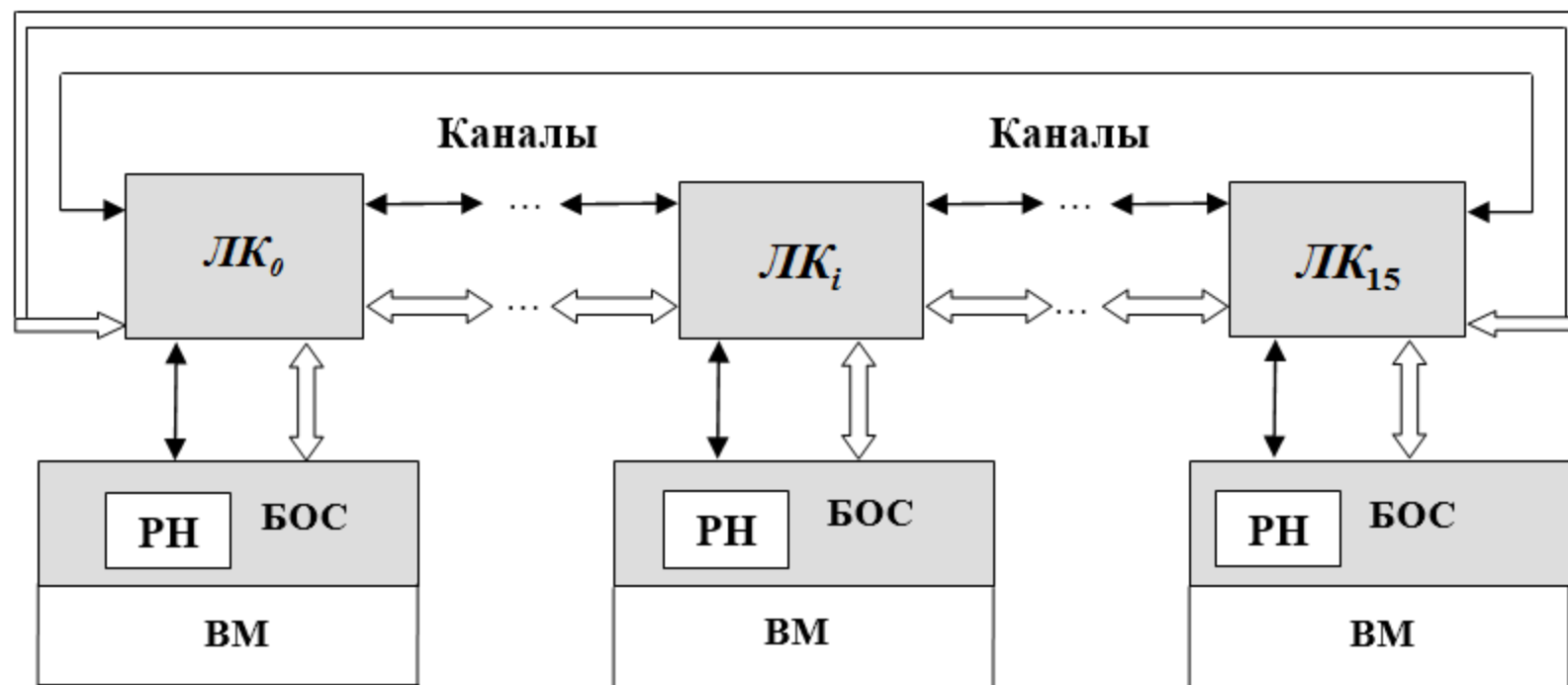
Частота - 250кГц

Система счисления - двоичная с плавающей и фиксированной запятой

Потребляемая мощность - 30кВт

Площадь - 240м²

Функциональная структура «Минск-222»



⇨ — рабочий канал, —————> — управляющий канал,

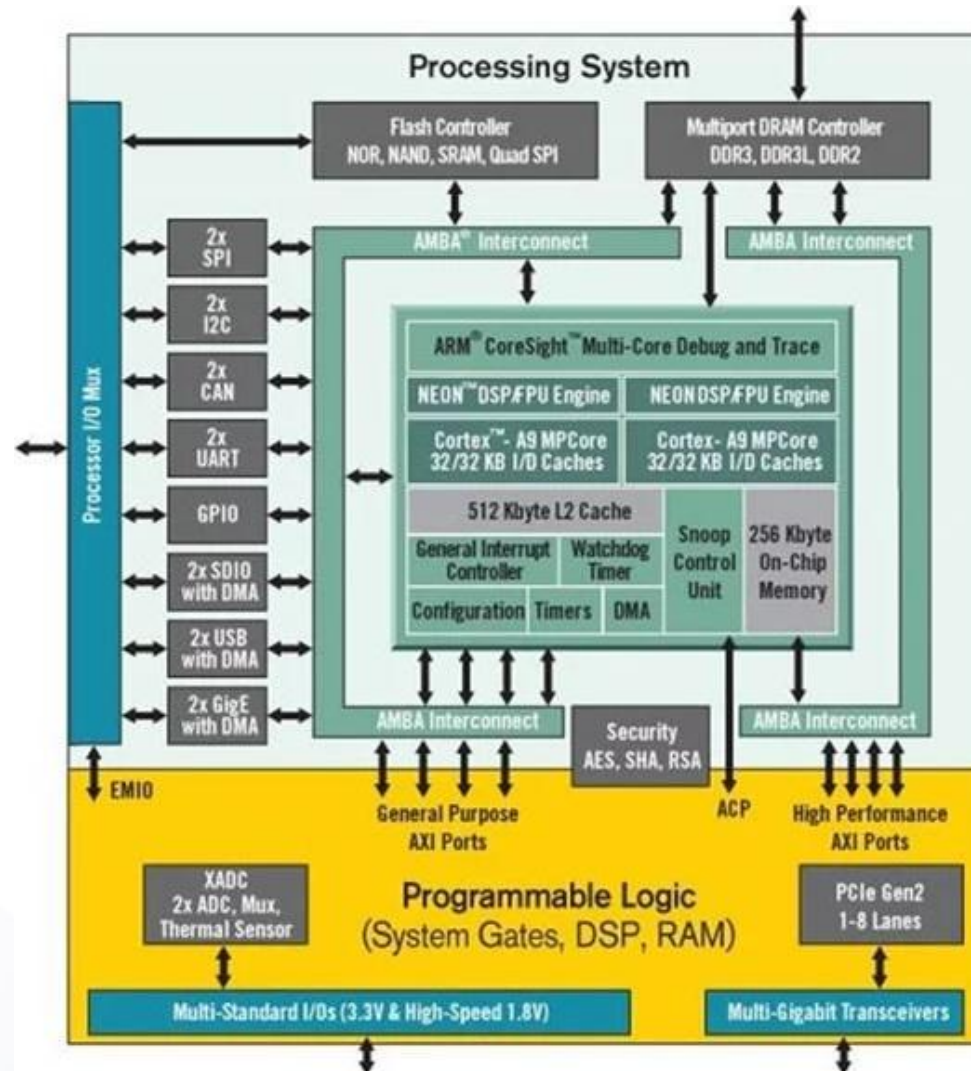
ЛК — локальный коммутатор, БОС — блок операций системы, РН — регистр настройки,
ВМ — вычислительный модуль

Современные ВС с программируемой структурой

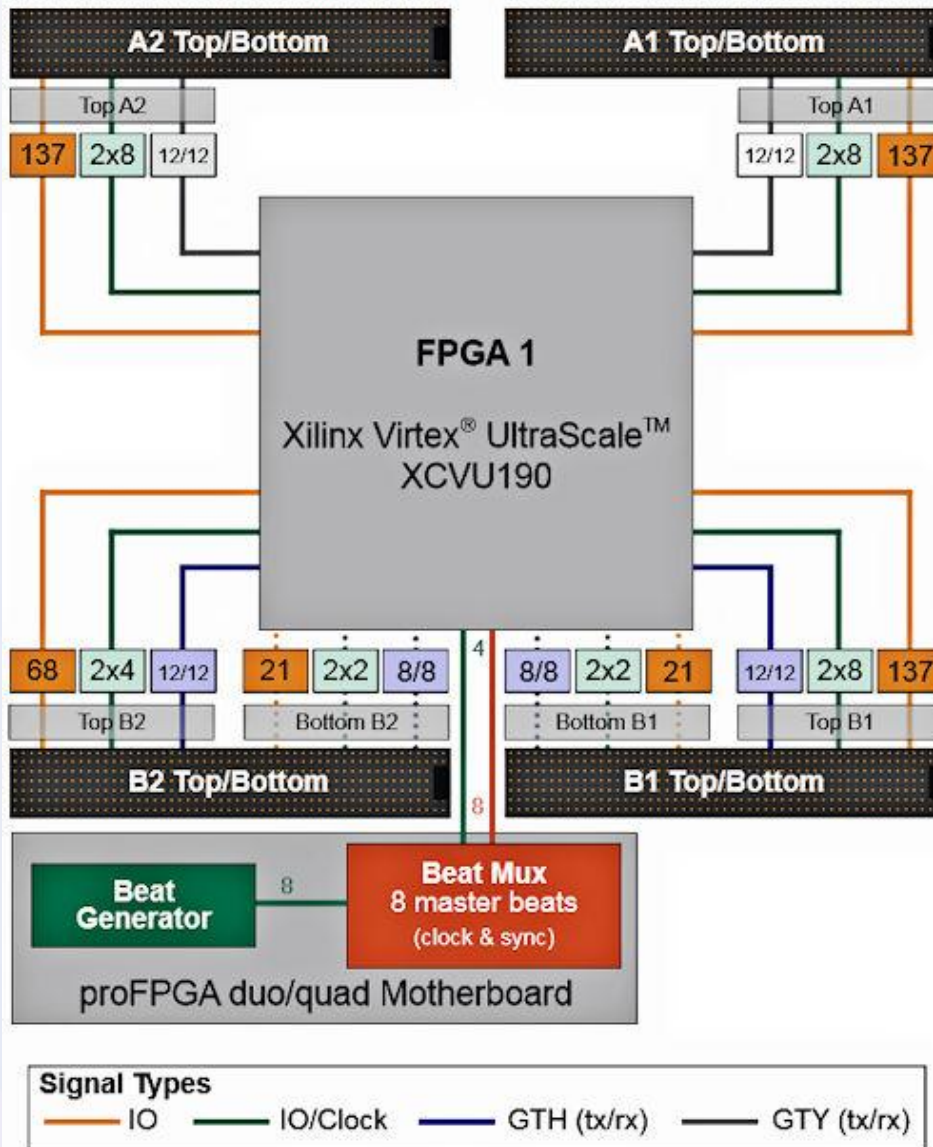
Программируемая пользователем вентильная матрица (ППВМ, англ. field-programmable gate array, FPGA) — полупроводниковое устройство, которое может быть сконфигурировано производителем или разработчиком после изготовления; наиболее сложная по организации разновидность программируемых логических интегральных схем.

VHDL (англ. VHSIC (Very high speed integrated circuits) Hardware Description Language) — язык описания аппаратуры интегральных схем. Язык проектирования VHDL является базовым языком при разработке аппаратуры современных вычислительных систем.

Гибридные узлы с программируемой структурой



Xilinx UltraScale Architecture

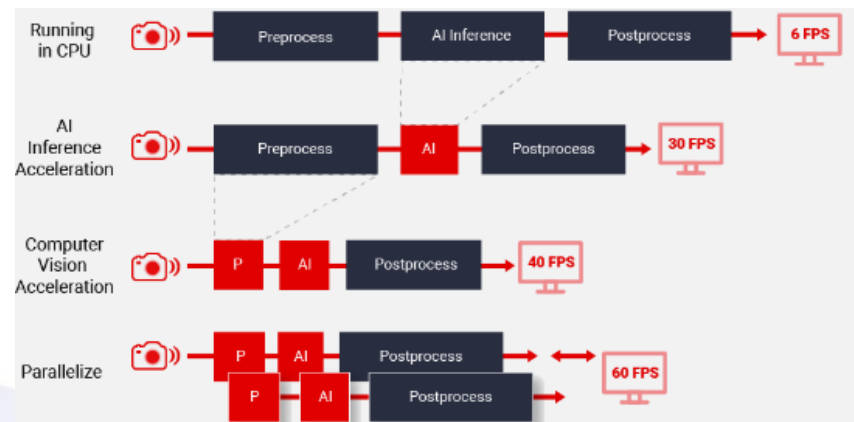
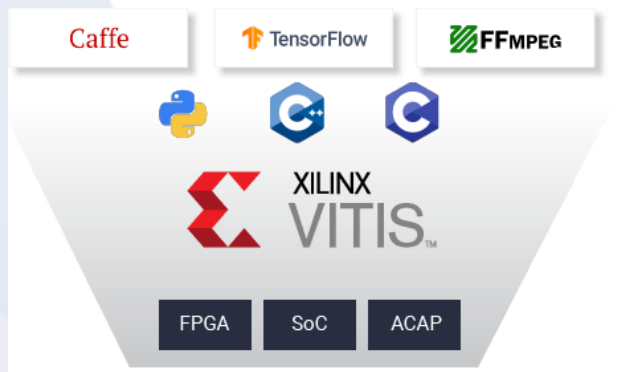


Язык высокого уровня COLAMO

Язык высокого уровня с неявным описанием параллелизма COLAMO является новым языком для программирования реконфигурируемых систем и позволяет программисту максимально просто описывать различные виды параллелизма в достаточно сжатом виде. Язык COLAMO содержит конструкции, аналогичные конструкциям традиционных языков программирования высокого уровня, но, в то же время, позволяет в едином контексте описать как структурную, так и процедурную компоненты параллельной программы. Языковые средства оперируют традиционными для большинства систем программирования операторами условий, циклов и арифметических операций, представленными в виде конструкций, аналогичных синтаксису языков Паскаль и Фортран.

XILINX VITIS Unified software platform

- The Vitis unified software platform enables the development of embedded software and accelerated applications on heterogeneous Xilinx platforms including FPGAs, SoCs, and Versal ACAPs. It provides a unified programming model for accelerating Edge, Cloud, and Hybrid computing applications.
- Leverage integration with high-level frameworks, develop in C, C++, or Python using accelerated libraries or use RTL-based accelerators & low-level runtime APIs for more fine-grained control over implementation — Choose the level of abstraction you need.



Литература

Хорошевский В.Г. Архитектура вычислительных систем. Учебное пособие. – М.: МГТУ им. Н.Э. Баумана, 2005; 2-е издание, 2008.

Хорошевский В.Г. Инженерные анализ функционирования вычислительных машин и систем. – М.: “Радио и связь”, 1987.